Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 33 (2015) 99 – 104

www.elsevier.com/locate/procedia

9th CIRP Conference on Intelligent Computation in Manufacturing Engineering - CIRP ICME '14

Open-Loop-Feedback Scheduling for Multi-Stage Production Systems

Agostino Villa[a], Teresa Taurino[a]*

Politecnico di Torino, corso Duca degli Abruzzi, 24, 10129, Torino, Italy

* Corresponding author. Tel.: +39 011-090.7234; fax: +39 011-090.7299. *E-mail address:* teresa.taurino@polito.it

Abstract

The task of scheduling product orders and updating the schedule at each arrival of a new order, especially if received from an “important” client, is of crucial importance in Small-Medium Enterprises - SME. These firms, in fact, are often operating according to production-to-order with a variety of products whose amplitude depends on the industry. This note proposes a new procedure for dynamic scheduling, that means re- scheduling at the occurrence of a new job to be produced. The re-scheduling procedure is specifically developed for manufacturing SMEs, in which each job is defined: i) by a precise machining sequence that must be followed exactly; ii) by a date at which orders are released together with the necessary raw materials, and especially iii) by a delivery date that must be satisfied. Accounting for these strong constraints, from the theoretical point of view, the solution of the dynamic scheduling problem is obtained by applying the constraint programming approach, with an innovative formulation of matrix type. The manufacturing system is represented in terms of a multi-stage model, able to process a number of different products orders, each one to be manufactured according a specific “path”, i.e. a sequence of work centers. Each time a new order will arrive to the system, an upgrading of the schedule has to be done, and no assumption about the next arrival time neither the next order type and dimension can be done.

As the need for a simple but robust procedure to re-scheduling is especially urgent in the field of footwear production, which contributes significantly to the export of "made in Italy", the case study - to test the procedure - is derived from data of an FMS belonging to this sector, operating in southern Italy.

Selection and peer-review under responsibility of the International Scientific Committee of “9th CIRP ICME Conference”

Keywords: Dynamic sceduling; multi-stage production system.

1. Introduction

To generate and frequently update production orders’ schedules is the crucial organizational task that must be approached in any production system, such as to plan the best utilization of manufacturing facilities and to assure the respect of orders’ delivery (due) dates as well as raw materials release times. Any production system is continuously driven by arrivals of new orders (often, unexpected “events”): then, a dynamic upgrading of schedules is mandatory so that the system could react quickly to new events. Re-scheduling can also be forced by the occurrence of some disturbances, as either events affecting the work centers (e.g. casual failures that could affect both the processing rate and the product quality) or events due to late materials supply. However, while failures have to be avoided as much as possible by preventive maintenance and material supply has to be carefully planned by the production flows management, re-scheduling of production orders must be performed through a proper procedure. If we take into account the studies on the problem of dynamic scheduling, we can find an extensive literature, as also described in some survey papers [1][2], where the two main aspects of the problem of re-scheduling in the presence of events are summarized. The first aspect concerns two questions: when to re-schedule and how to react to events. The second aspect is related to what techniques to use in dynamic scheduling. With reference to the first aspect, three types of decisions can be taken when apply a re-schedule: either a periodic re-schedule, or an event-driven or an hybrid. In the first case, a re-schedule is generated at fixed time buckets; in the second case, re-scheduling is applied in front of any event; in the third case, usual re-planning occurs periodically and

Selection and peer-review under responsibility of the International Scientific Committee of “9th CIRP ICME Conference”
doi:10.1016/j.procir.2015.06.019

sometime in response to an events (mainly if of prevailing impact).

Dynamic scheduling techniques have been classified either according to the rules to be applied, or the way of applying the schedule update in the different work centers [3]. So one can find techniques based on dispatching rules, heuristics, and also artificial intelligence techniques and multi-agent structures (a comprehensive survey in [2]). The most recent development of methods for re-scheduling is aimed at the development of techniques that require the application of multi-agent-based architectures. In this way, the agents correspond to machining centers or islands, and each center has its partial autonomy to define their own local schedule. Alternatively, a multi-agent approach is split as an agent with the individual job autonomy, which is allowed to choose their own "road" in the network of processing centers, almost from the perspective of a dynamic programming of its path [4, 5]. The problem with these methods is that their application makes sense in the case of complex manufacturing systems, consisting of many processing centers or manufacturing cells, with several alternatives for their utilization (eg, flexible and multi-product centers and islands). There are not considered in the development of such techniques Small Mid Enterprises - SME, which all have instead the following constraints: a few different manufacturing/work sequences (ie, the sequence of manufacturing operations for anyone of the few families of final products) and with little variation over time, tight constraints for both the release of the materials by suppliers, and especially for delivery dates of final products to customers.

Case study:
As a typical example of the need for a dynamic scheduling procedure, the case of the manufacturing SME producing shoes of four different types is considered: namely, accident prevention shoes, shoes for women and children, slippers, and rubber shoes. This SME, located in a small town in South Italy, employs 32 people to manufacture, has annual sales of about 2.30 million Euro and exposes a percentage of exports amounting to over 55%. Whether as a product that offered as internal organization, this small business can be placed between the small Italian excellences. The work sequences necessary for the four types of final products can be divided in two stages. The first working stage includes the activity of shearing of the soles, the manufacturing of heels and trenching of uppers; the second working stage includes the activities of sewing, gluing and soles injection.

For what concerns the scheduling needs, the SME management calls for a re-scheduling techniques able [6-8]:

a) to take into account the few number of work sequences, one for each type of final product;
b) to apply an upgrading of the schedule each time a new order is released,
   - such to satisfy both release times and due dates of all orders to be re-scheduled,
   - but such that no operation but such that no shift is applied with every operation already started in a earlier time compared to that of re-scheduling.

Based on these industrial specifications (that are typical for any manufacturing SME, also in different sectors, as noted in the experience done by authors in the European project CODESNET – Collaborative Demand and Supply NETworks www.codesnet.polito.it), the following Section will present a formulation of the dynamic scheduling problem including all constraints above mentioned.

2. A dynamic scheduling model formulation

The terminology used in this paper is as follows.

Nomenclature

n	number of lots
m	number of machines
t_e	arrival time of the e-th event E
l_{ij}	processing time for job i on machine j
R_i	release time of job i
D_i	due date of job i
x_{ij}	starting time of processing lot i on machine j
sl_i	slack time for lot i
a_{ij}	waiting time of lot i on machine j
b_{ij}	idle time of machine j before the i-th operation
π_{ij}	position of machine j in the working sequence of lot i
η_{ij}	position of lot i in the sorting on machine j
P_i	permutation matrix representing the working sequence for lot i
Q_j	permutation matrix representing the sequence of lots on machine j
Π	matrix of π_{ij} elements
H	matrix of η_{ij} elements
A	matrix of a_{ij} elements
B	matrix of b_{ij} elements
$S_j(t_e)$	set of lots to be rescheduled at time t_e
s_j	cardinality of set $S_j(t_e)$
K	set of new lots
k	cardinality of the set K

2.1. The dynamic model

Let us consider n different lots to be scheduled on m machines with a *non-preemptive* scheduler. Let l_{ij} denote the processing time for lot i on the machine j, the matrix L is a $\mathfrak{R}^{n\times m}$ matrix of the processing times:

$$L = \{l_{ij}\}_{i=1,\dots,n.\ j=1,\dots,m} \tag{1}$$

Let r_i and d_i denote, respectively, the release time and the due date for lot i. Real vectors R and D have length n.

We are interested in finding the unknown variable of starting time x_{ij} for every lot i on the machine j. In matrix notation, the unknown variable is an n x m matrix such that:

$$X = \{x_{ij}\}_{i=1,\dots,n.\ j=1,\dots,m} \tag{2}$$

Let us introduce two permutation matrices P and Q.

The P matrix is a square m x m binary matrix that has exactly one element equal to 1 in each row and each column, while the other elements are equal to 0.

$$(P_i)_{jk} = \begin{cases} 1 & if\ k = \pi_{ij} \\ 0 & otherwise \end{cases} \tag{3}$$

Where π_{ij} is an element of matrix Π with the position of the machine j in the working sequence of lot i. This matrix represents a specific permutation of m elements and, by multiplying on the right the X matrix with the matrix P, we obtain a permutation of columns of X. Each column of X represents a machine, so the permutation of columns of X we are interested in are permutation according with the working sequence. In general, for each lot we have a different working sequence, so each lot i requires a permutation matrix P_i of the columns of X according to the working sequence of lot i.

The Q matrix is a square n x n permutation matrix of rows of X. The purpose of the Q matrix is to sort the n lots, to every machine j, according to the scheduled sequence on the machines. This is possible by multiplying on the left the matrix X with the permutation matrix Q. In general, on each machine we can have a different order, so the permutation matrix Q_j is the matrix that sort the rows of X according to the scheduled order on machine j.

$$(Q_j)_{ik} = \begin{cases} 1 & if\ k = \eta_{ij} \\ 0 & otherwise \end{cases} \tag{4}$$

Where η_{ij} is an element of the matrix H with the position of lot i in the sorting on machine j. Let us assume that the set-up times are negligible for each machine and for each sorted couple of lots. Let us consider the e-th event E that generates the requirement of a (re-)scheduling of the lots. The event E occurs at time t_e and can be:

- the arrival of k ($k \geq 1$) new lots;
- changing on the working sequences;
- changing on due dates.

Let us consider the first kind of event. And let K be the set of the new lots that arrives at time t_e:

$$K = \{n+1, ..., n+k\} \tag{5}$$

For the last two kinds of re-scheduling events, the procedure is obtained considering $K=\emptyset$. About the new k lots, for all j=1,…,m, we know the working sequences $\pi_{n+1\,j} \ldots \pi_{n+kj}$, the processing times $l_{n+1\,j} \ldots l_{n+kj}$, the release times $R_{n+1}, \ldots, R_{n+k}$. and the due dates $D_{n+1}, \ldots, D_{n+k}$. Let S_j be the set of lots to be re-scheduled on machine j because the scheduled starting time of the lot i on machine j is subsequent to the event time t_e.

$$S_j(t_e) = \{i : x_{ij} \geq t_e\} \tag{6}$$

with a cardinality equal to s_j ($s_j \leq n$). The dynamic scheduling model is in Eq.(7):

$$\min_{Q_j} \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij}(Q_j) \tag{7}$$

$$s.t.\ (XP_i)_{i,1} \geq R_i \qquad i \in \bigcup_j S_j(t_e) \cup K \tag{8}$$

$$(Q_j XP_i)_{i,j} \geq (Q_j XP_i)_{i,j-1} + (Q_j LP_i)_{i,j-1} \quad with \quad i \in S_j(t_e) \cup K \tag{9}$$

$$(Q_j XP_i)_{i,j} \geq (Q_j XP_i)_{i-1,j} + (Q_j LP_i)_{i-1,j} \quad with \quad i \in S_j(t_e) \cup K \tag{10}$$

$$(XP_i)_{i,m} + (LP_i)_{i,m} \leq D_i \qquad i \in \bigcup_j S_j(t_e) \cup K \tag{11}$$

$$S_j(t_{e+1}) = \{i : x_{ij} \geq t_{e+1}\} = S_j(t_e) \setminus \{i : t_e \leq x_{ij} < t_{e+1}\} \tag{12}$$

where a_{ij} is the waiting time of lot i for working operation j.

$$a_{ij} = (XP_i)_{i,j} - (XP_i)_{i-1,j} - (LP_i)_{i-1,j} \tag{13}$$

Eq.(8) states that each lot can start the first working operation after it has been released, i.e. the start time of the first working operation must be greater or equal to the release time (*constraint of release*). The number of constraints defined in Eq.(8) is equal to s_i+k. Eq.(9) states that, for each lot, a working operation can start if the previous working operation, in the working sequence, has been completed *("constraint of precedence")*. In other words, the difference between the starting time of two consecutive operations must be equal or greater than the working time of the first of the two operations. The number of constraints defined in Eq.(9) is equal to m-1. Eq.(10) represents the constraint of *"not overlapping"*: each machine can process only one lot a time, i.e. given the sorting of lots on each machine, the processing of next lot can start after the completion of the working operation on the previous lot. The number of constraints defined in Eq.(10) is equal to s_i+k-1. Eq.(11) introduces the *constraint of compliance with the due dates*, the constraint is on the last working operation in the working sequence that, thanks to the permutation matrix P, is, for each lot i, in the m position in the XP_i and LP_i matrices. The number of constraints defined in Eq.(11) is equal to s_i+k. Eq.(12) represents the dynamic variation on the sets of lots that have to be scheduled on each working operation; the sets are time dependent from the arrival of a new event of re-scheduling at time t_{e+1}. By knowing the matrix X with the starting times of operations, the matrix B with the idle times on each machine is defined with the following equation:

$$b_{ij} = (Q_j X)_{i,j} - (Q_j X)_{i-1,j} - (Q_j L)_{i-1,j} \tag{14}$$

2.2. The algorithm

Before to start with the re-scheduling algorithm, a control on the consistency of the due dates is required in order to be sure to have at least one admissible solution. The due dates must be consistent with the release times and the processing times in order to define a region with an admissible solution.

Let us introduce the slack time for lot i as in Eq.(15):

$$sl_i = D_i - R_i - \sum_{j=1}^{m} l_{ij} \qquad \forall i = 1,\ldots,s_j + k \tag{15}$$

The slack time, in general, is a non-negative number. To be sure to have at least one solution with all the jobs on time within their due dates, the slack times must respect the following condition:

$$sl_i \ge \min_{Q_j} \sum_{j=1}^{m} a_{ij}\left(Q_j\right) \qquad \forall i = 1,\ldots,s_j + k \tag{16}$$

This waiting times a_{ij} are function of the sorting of lots on machine j. If we choose, as first sorting of lots, on each machine, the release order, we can find a compliant set of slack times even if not the set with the minimum value of slack times.

The algorithm can be written as follows:

Step 0: Verifying the existence of an admissible solution;

Step 1: Apply the scheduling procedure.

Step1.1: Assign to each machine the release times order of lots. In this way the constraint in Eq.(8) is verified.

Step 1.2: Forward propagation with the computation of matrix X, respecting, in the construction constraints in Eq.(9-10)

Step 1.3: Verifying the respect of the constraint on the due dates Eq.(11) (the other constraints – Eq.(9-10) - are respected for construction of the solution X).

- If the due date is not respected:

Step 1.3.1.1: Consider the lot in late.

Step 1.3.1.2: Commute with the previous lot on each machine.

Step 1.3.1.3: Go back to *Step 1.2.*

- If the due date is respected:

Step 1.3.2: Reduce the waiting times a_{ij}.

Step 1.3.2.1: Select (i, j) such that $a_{ij}>0$.

Step 1.3.2.2a: Commute the lot with the next lot

Or

Step 1.3.2.2b: Commute the lot with the previous lot if this has not been rescheduled in *Step 1.3.1* to respect its constraint on the due date.

Step 1.3.2.3: Go back to *Step 1.2.*

Step 2: Rescheduling.

Step 3.1: Recognize the arrival of a new event E.

Step 3.2: Update the value of S_j for all j=1,…,m.

Step 3.2: Go back to *Step 1.*

This algorithm helps the constrain programming solver to find a solution of this complex problem [9, 10].

3. Application of the scheduling algorithm to the case study

Let us consider a scheduling problem of 4 lots on 2 working stages with the working sequence for each lot defined in Table 1.

Table 1. Working sequences (π_{ij})

	working stage WS1	working stage WS2
Lot 1	1	2
Lot 2	1	2
Lot 3	2	1
Lot 4	2	1

The processing time (in time unit) for the operations of the lots on the two working stages, the release times and the due dates are defined in Table 2

Table 2. Processing times (l_{ij}), Release times (R_i) and Due Dates (D_i)

	WS1 (l_{i1})	WS2 (l_{i2})	R_i	D_i
Lot 1	5	3	5	∞
Lot 2	2	1	3	∞
Lot 3	3	5	8	∞
Lot 4	2	2	6	10

The working sequence in Table 1 defines the permutation matrix P_i with $i=1,\ldots,n$ as in Eq.(3). In our example n=4 and m=2, so the four 2x2 matrices P_i are:

$$P_1 = P_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad P_3 = P_4 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{17}$$

As first tentative of sorting lots on the working stages let us consider lots in an increasing release times order as in Table 4:

Table 4. values of η_{ij}

WS1	WS2
2	2
1	1
4	4
3	3

Since the first sorting of lots is the same for both working stages, the matrices Q_1 and Q_2, derived from Eq.(4), are:

$$Q_1 = Q_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{18}$$

The initialization of matrix X is an n x m matrix with all elements equal to 0 except the element x_{ij}, where j is the first working stage in the working sequence of lot i, equal to the release time R_i. The result is a matrix that respects the constraint in Eq.(6):

$$X = \begin{bmatrix} 5 & 0 \\ 3 & 0 \\ 0 & 8 \\ 0 & 6 \end{bmatrix} \quad (19)$$

With two cycles on i=1,…,n and j=1,…,m, we calculate all the elements of X verifying the constraints in Eq.(9-10). From the matrix X it's possible to calculate the matrix A of the waiting times and the matrix B of the idle times (Eq.(13-14)).

$$X = \begin{bmatrix} 5 & 10 \\ 3 & 5 \\ 20 & 15 \\ 15 & 13 \end{bmatrix} \quad A = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 7 \\ 0 & 7 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 4 \\ 5 & 0 \\ 3 & 0 \end{bmatrix} \quad (20)$$

The constraint on the due dates in Eq.(6) is not verified, the completion time for lot 4 is 7 time units after the due date. Looking at matrix A, for lot 4, there is a waiting time of 7 time units on working stage 2. It's possible to find a permutation of lots on working stage 2 in order to verify the due date constraint. Matrix B put in evidence an idle time on the working stage 2 between the first and the second lot (lot 2 and lot 1), so, putting lot 4 between lot 2 and lot 1, on working stage 2 we obtain new values for η_{ij}:

Table 5. new values of η_{ij}

WS1	WS2
2	2
1	4
4	1
3	3

By computing the matrices X, A and B, respectively of starting times, of waiting times and of idle times, we obtain:

$$X = \begin{bmatrix} 5 & 10 \\ 3 & 5 \\ 18 & 13 \\ 10 & 6 \end{bmatrix} \quad A = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 5 \\ 2 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 2 \\ 6 & 0 \end{bmatrix} \quad (21)$$

The constraint on the due dates in Eq.(6) is not verified, the completion time for lot 4 is 12, 2 time units after the due date of 10. Looking to the A matrix, for lot 4, there is a waiting time of 2 time units on working stage 1. Since the delay on the due date is less or equal to the waiting time, it's possible to find a permutation of lots on working stage 1 in order to verify the due date constraint. Matrix B put in evidence an idle time on the working stage 1 between the third and the fourth lot (lot 4 and lot 3), this idle time occurs after the operation on lot 4. Changing the position, on working stage 1, permuting lot 4 with lot 1, the new values for η_{ij} are:

Table 6. new values of η_{ij}

WS1	WS2
2	2
4	4
1	1
3	3

By computing the matrices X, A and B, respectively of starting times, of waiting times and of idle times, we obtain:

$$X = \begin{bmatrix} 10 & 15 \\ 3 & 5 \\ 23 & 18 \\ 8 & 6 \end{bmatrix} \quad A = \begin{bmatrix} 5 & 0 \\ 0 & 0 \\ 0 & 10 \\ 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 3 & 0 \\ 0 & 7 \\ 8 & 0 \end{bmatrix} \quad (22)$$

The completion time of lot 4 is compliant with the due date, so the objective to respect the due dates is reached. We have found an admissible solution. We can improve the solution reducing the waiting of lots. In particular there are two lots with non-zero waiting times: lot 1 and lot 3. With the constraint that we cannot downgrade the position of lot 4, we can try a permutation of lots 1 and 3 on the working stages.

Table 7. new values of η_{ij}

WS1	WS2
2	2
4	4
3	3
1	1

By computing the matrices X, A and B, respectively of starting times, of waiting times and of idle times, we obtain:

$$X = \begin{bmatrix} 16 & 21 \\ 3 & 5 \\ 13 & 8 \\ 8 & 6 \end{bmatrix} \quad A = \begin{bmatrix} 11 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 3 & 0 \\ 3 & 0 \\ 0 & 8 \end{bmatrix} \quad (23)$$

Only lot 1 has to wait for the working operation on working stage 1, so we can stop our algorithm and take Eq.(23) as the final solution of the problem.

Let us consider now the arrival of a new lot 5 at time R_5=7 with a due date D_5=19 and processing times l_{51}=1 and l_{52}=6. The working sequence for lot 5 is working stage 1 and then working stage 2. We need to compute again the scheduling problem starting from the previous solution. We cannot change working operation already started at the arrival of the new job, so we cannot change the scheduling of lot 2 on the working stage 1 and of lots 2 and 4 on the working stage 2.

For the other couple (lot, working stage) the rule in *Step 1.1* of the algorithm, is to follow the order given by the release time, so the matrix H is:

Table 8. new values of η_{ij}

WS1	WS2
2	2
1	4
4	1
5	5
3	3

By following the algorithm we obtain the matrices *X, A, B:*

$$X = \begin{bmatrix} 5 & 10 \\ 3 & 5 \\ 24 & 19 \\ 10 & 6 \\ 12 & 13 \end{bmatrix} \quad A = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 11 \\ 2 & 0 \\ 5 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 2 \\ 0 & 0 \\ 11 & 0 \end{bmatrix} \tag{24}$$

This solution doesn't respect the constraint on the due date for lot 4, so let us permute the lot on working stages 2.

Table 9. new values of η_{ij}

WS1	WS2
2	2
4	4
1	1
5	5
3	3

By following the algorithm we obtain the matrices *X, A*, *B*:

$$X = \begin{bmatrix} 10 & 15 \\ 3 & 5 \\ 29 & 24 \\ 8 & 6 \\ 15 & 18 \end{bmatrix} \quad A = \begin{bmatrix} 5 & 0 \\ 0 & 0 \\ 0 & 16 \\ 0 & 0 \\ 8 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 3 & 0 \\ 0 & 7 \\ 0 & 0 \\ 13 & 0 \end{bmatrix} \tag{25}$$

This solution is compliant with the constraint on the due date for lot 4, but not for lot 5. So we need to change the position of lot 5 on the working stage 1 using a branch and bound approach with the previous lot.
This solution wouldn't be compliant with the constraint on the due date for lot 5. So we need to change the position of lot 5 also on the working stage 2:

Table 11. new values of η_{ij}

WS1	WS2
2	2
4	4
5	5
1	1
3	3

By computing the matrices *X*, *A* and *B*, we obtain:

$$X = \begin{bmatrix} 11 & 17 \\ 3 & 5 \\ 25 & 20 \\ 8 & 6 \\ 10 & 11 \end{bmatrix} \quad A = \begin{bmatrix} 6 & 0 \\ 0 & 0 \\ 0 & 12 \\ 0 & 0 \\ 3 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 3 & 0 \\ 0 & 3 \\ 0 & 0 \\ 9 & 0 \end{bmatrix} \tag{26}$$

This solution satisfies all the constraints in Eq.(8-11)

4. Some scheduling remarks

The dynamic scheduling method presented in the paper appears to be particularly interesting for a manufacturing SME characterized by the typical constraints on orders' release time and due dates that big enterprises, acting as SME client, apply to the receipt of their products. The developed re-scheduling method is characterized by the following properties:

a) It always respects the working sequence, that is the technological constraint for manufacturing any job;
b) It satisfies both release time and due date constraints, that are the strongest constraints on SME operations management;
c) It applies a few "branch-and-bound" steps, to improve the scheduling solution at each re-scheduling event, thus reaching a better solution than a dispatching approach.

The application to the SME operating in the field of footwear production, gives an evident view of the possibility of easy application of the method.

References

[1] Vieira G. E., Herrmann J. W., Lin E. Rescheduling manufacturing systems: A framework of strategies, policies and methods. J. Scheduling 2003; 6:39-62.

[2] Quelhadj D., Petrovic S. Survey of dynamic scheduling in manufacturing systems. J. Scheduling. 2009; 12: 4: 417-431.

[3] Brandimarte P., Villa A. Advanced Models for Manufacturing Systems Management, Boca Raton: CRC Press, Mathematical Modelling Series; 1995.

[4] Merdan, M., Vrba, P., Koppensteiner, G., Zoitl, A. Knowledge-based multi-agent architecture for dynamic scheduling in manufacturing systems. 6th IEEE International Conference on Industrial Informatics, 2008. INDIN 2008. Daejeon; p. 1075-1080.

[5] Shena W., Haoa Q., Yoona H.J., Norrieb D.H. Applications of agent-based systems in intelligent manufacturing: An updated review. Advanced Engineering Informatics; 2006; 20:4: 415–431.

[6] Marinho, J., Braganca, A., Ramos, C. Proc. of the 1999 IEEE International Symposium on Decision support system for dynamic production scheduling. Assembly and Task Planning, 1999. (ISATP '99). Porto. 1999, p. 424-429.

[7] Davorin Kofjač D., KnafličA., Kljajic M. Development of a Web Application for Dynamic Production Scheduling in Small and Medium Enterprises. Organizacija. 2010; 43: 3: 125–135. [8] Hutzschenreuter J. Management Control in Small and Medium-Sized Enterprises. PDF Books Planet; 2014.

[9] Kelbel Z., Hanzálek Z. Solving production scheduling with earliness /tardiness penalties by constraint programming. J. Intelligent Manufacturing. 2011; 22;4:553-562.

[10] Van Hentenryck P., Laurent Michel L. Constraint-Based Local Search. The MIT Press; 2009